



Working to Perfect the Flow of Energy

NSI Data Specification Guide

by PJM Interconnection

Version 1.0

Prepared by
Transmission Service Department

© PJM 2020

Table of Contents

| | |
|--|----|
| 1. Introduction | 3 |
| 1.1. Purpose | 3 |
| 1.2. Scope..... | 3 |
| 1.3. Acronyms..... | 3 |
| 1.4. Regulatory References..... | 3 |
| 1.5. Other References..... | 3 |
| 2. Data Conventions | 4 |
| 2.1. Date Format Convention..... | 4 |
| 2.2. Creator BA field..... | 4 |
| 2.3. Requestor BA field | 4 |
| 2.4. Checkout BA field | 4 |
| 2.5. Sink BA field | 5 |
| 3. Data Transfers Overview | 6 |
| 3.1. Request-Response..... | 6 |
| 4. Building a Request | 7 |
| 4.1. Request Parameters..... | 7 |
| 5. Building a Payload | 9 |
| 5.1. Request Metadata..... | 9 |
| 5.2. Real-Time NSI..... | 9 |
| 5.3. Daily NSI | 10 |
| 5.4. Optional Integrated Interval Data | 10 |
| 5.5. Optional eTag Data..... | 10 |
| 6. Three-Part Verification | 13 |
| 7. XSD Definitions | 16 |
| 7.1. Schema..... | 16 |
| 7.2. Sample XML | 18 |
| 8. Terms of Use | 21 |

Revision History

| Date | Version | Change |
|------------|---------|-------------------|
| 09/30/2019 | 0.1 | Initial Draft |
| 07/09/2020 | 1.0 | Legal terms added |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

1. Introduction

1.1. Purpose

The purpose of this document is to describe the architecture and design of the services required to exchange Net Scheduled Interchange (NSI) data for the purpose of automating NSI checkout between two Balancing Authorities (BA). BAs can use the NSI data received from a neighboring BA to verify the NSI calculated by their internal interchange software, eliminating the need for a phone call. In addition to NSI data, data exchange partners would have the liberty to develop and include customized, non-standard data elements as needed.

1.2. Scope

The scope of this document is limited to the proposed architecture and design of data transfers required for automating NSI checkout between two BAs. This document is intended to provide specifications for the development and delivery of all data elements needed to support this automation. While this document does outline a detailed approach, the specified transfers and attributes may be altered during joint design between two BAs throughout the course of developing a formal data exchange tool.

1.3. Acronyms

| Term | Meaning |
|------|---------------------------------|
| ATF | After the fact |
| BA | Balancing Authority |
| BTF | Before the fact |
| CCI | Composite Confirmed Interchange |
| NSI | Net Scheduled Interchange |
| RT | Real Time |
| UTC | Coordinated Universal Time |

1.4. Regulatory References

NERC INT-009-2.1 R1. 1.1, 1.2.

1.5. Other References

The most recent copy of the Automated Checkout Data Specification can be found at <https://pjm.com/markets-and-operations/etools/exschedule/ansi-automated-checkout.aspx>

The most recent copy of the Automated Checkout XML Schema can be found at <https://pjm.com/-/media/etools/exschedule/ansi-xsd.ashx?la=en>

For detailed information regarding NERC Standards, please see <https://www.nerc.com/>

2. Data Conventions

2.1. Date Format Convention

All interactions use a time interval. The data type of **dateTime** is used to specify a date and a time. The startDate and endDate fields are specified in ISO format, which is defined as "YYYY-MM-DDThh:mm:ss(Z or ±-hh:mm)" where:

| Field | Description |
|--------------|--|
| YYYY | Indicates the year |
| MM | Indicates the month |
| DD | Indicates the date |
| T | Indicates the start of the required time section |
| hh | Indicates the hour |
| mm | Indicates the minute |
| ss | Indicates the second |
| Z or ± hh:mm | UTC 'Zulu' time or timezone offset |

Example A:

2019-09-30T16:00:00.000Z

Example B:

2019-09-30T12:00:00-04:00

2.2. Creator BA field

The creatorBA field should be filled in with the registered Entity Code of the BA who is responding to the data request and creating a response payload.

2.3. Requestor BA field

The requestorBA field is used to specify the list of neighboring BAs for whom the payload is created. This field can be derived from the values supplied in the 'area' parameter.

This is an optional, unbounded field. It is suggested to fill this field with the registered Entity Code of the requestor BA. The creator of the payload would determine who the requestor is when the request passes authentication. If the requestor is authorized to receive NSI data for more than one area, then this element will be repeated with multiple values.

2.4. Checkout BA field

The checkoutBA field is nested under complexType of NsiTotals and DailyNsiTotals. Both complexTypes are unbounded so they can be used for sending NSI data to multiple areas within a single payload.

The requestor BA may send a comma separated list with multiple areas' Entity Code. When responding to a request with multiple areas, the responder should use **one** checkoutBA for each BA NsiTotal.

Example:

Example of the response payload to the request with comma separated list of checkoutBA field.

```

<NsiTotals>
  <NsiTotal>
    < checkoutBA>CPLE</ checkoutBA>
    <NsiIntervals>...</NsiIntervals>
    <IntegratedIntervals>...</IntegratedIntervals>
  </NsiTotal>
</NsiTotal>

```

```

<checkoutBA>CPLW</checkoutBA>
<NsiIntervals>...</NsiIntervals>
<IntegratedIntervals>...</IntegratedIntervals>
</NsiTotal>
</NsiTotals>

```

2.5. Sink BA field

The sinkBA field is used in combination with the creatorBA field to show the direction of net energy flow. As a creator BA, when calculating NSI for an adjacent BA, it is important to isolate the tag’s path to only two BAs: creator BA and the adjacent requestor BA.

Example:

| # | Scenario | requestorBA | creatorBA | sinkBA | mwNet |
|---|---|-------------|-----------|--------|-------|
| 1 | For net energy flow of 200 MW from BA1 to BA2 | BA2 | BA1 | BA2 | 200 |
| 2 | For net energy flow of 500 MW from BA2 to BA1 | BA2 | BA1 | BA1 | 500 |

3. Data Transfers Overview

3.1. Request-Response

Description:

A request-response method is recommended for this data exchange, but BAs have the liberty to choose an alternate data transfer method. Request-response is a message exchange pattern where a requestor sends a request message to a replier system that receives and processes the request, and ultimately returns a message in response. This is a simple, but powerful messaging pattern that allows two applications to have a two-way conversation with each other. This pattern is especially common in client-server architectures.

For simplicity, this pattern is typically implemented in a purely synchronous fashion, such as in web service calls over HTTP, which holds a connection open and waits until the response is delivered or the timeout period expires.

Example:

A requesting BA initiates the request to retrieve data *from* neighboring BA systems. The requesting BA is making a GET request via the RESTful service passing in the start and stop times in UTC format.

The neighboring BA system has an exposed RESTful service that allows for retrieval of NSI data based on the start and stop times within the service request. The neighbor BA returns NSI data for the interval(s) requested in XML format. This NSI data contains every eTag within the requesting BA's area spanning between the start and stop times defined within the request call. The creator BA does not check if the published data is new, old, or delta. If eTags are not available within the creator BA's system to contribute towards NSI for the requested interval, an empty list is returned to the requestor.

4. Building a Request

4.1. Request Parameters

Description:

Data requests are submitted with inclusive start & stop time windows. The requestor must specify the request type. All other optional data are treated as false and not included if not requested.

Data Elements:

| Request Parameter | Required | Sample Value | Description |
|-------------------|----------|--------------|---|
| start | Yes | 201706060500 | This is beginning of the time inclusive request window, specified with a date and time. This means if Requestor needs scheduled NSI starting from 1300, they will supply 1300 as start time. |
| stop | Yes | 201706060500 | This is the end of the inclusive request window, specified with a date and time. It is expected NSI is received for the 15 minute valid scheduling interval before this data-time as long as it's after the Start time parameter. |
| area | Yes | MISO, CPLE | This lets the requestor to specify the list of areas for which to request NSI totals. (i.e., If a BA need to request NSI totals for DUK, CPLE & CPLW, then they can request multiple areas within the URL separated by comma.) |
| type | Yes | DAY or RT | Provide a type of NSI data. (e.g., RT, DAY). Details of the response for each request type are specified in the 'Building Payload' section. |
| tag | No | t or f | Provides an option to request the underlying eTags that support a given NSI value. If true, all underlying tags used to determine NSI values for the provided start and stop window will be returned. More details are available in the 'Optional eTag Data' section. |
| integrated | No | t or f | Provides an option to request hourly integrated totals along with RT/DAY totals. |

URL Operators:

The question mark, ampersand, equals sign and comma are operators used in the syntax of query strings/URL variables.

| Operator | Sample Value | Description |
|----------|---|--|
| ? | e.g. ?variable=value e.g. ?start=201706060500 | The question mark identifies the beginning of the query string and must be placed at the end of the link, before the contents of the query string. |
| & | e.g. variable=value&variable2=value2 e.g. tag=t&integrated=f | The ampersand is used before each subsequent variable/value pair in the query string. |
| = | e.g. variable=value e.g. tag=t | The equals sign separates the variable from the value assigned to that variable. |
| , | e.g. variable=value1,value2,value3 e.g. area=DUK,CPLW,CPLW | The comma is used to separate multiple values for a single variable. |

Date format for start & stop:

The parameters <start> and <end> dateTime must be UTC and follow the format YYYYMMDDhhmm, where:

- YYYY** = four-digit year
- MM** = two-digit month (01=January, etc.)
- DD** = two-digit day of month (01 through 31)
- hh** = two digits of hour (00 through 23) (am/pm NOT allowed)
- mm** = two digits of minute (00 through 59)

Example URL:

<https://nsi.pjm.com/NSI/rest/getnsi?start=201706060500&stop=201706060500&type=RT&tag=t&integrated=f&area=DUK,CPLW,CPLW>

5. Building a Payload

5.1. Request Metadata

Description:

Every response payload shall contain data to indicate the time window requested, type of data, and other parameters as requested. Even though this data is not used directly in the checkout calculation, it is included in human-readable format for troubleshooting purposes.

Data Elements:

| Field Name | Required | Sample Value | Description |
|-------------------|----------|--------------------------|--|
| requestStartTime | Yes | 2020-01-15T19:00:00.000Z | Fill this field with the value provided in the URL start request parameter. |
| requestStopTime | Yes | 2020-01-15T21:00:00.000Z | Fill this field with the value provided in the URL stop request parameter. |
| responseTimestamp | Yes | 2020-01-15T18:22:00.000Z | Fill this field with the time when the creator BA created this payload. |
| requestType | Yes | DAY or RT | Fill this field with the value provided in the URL type request parameter. |
| includeIntegrated | Yes | true or false | Fill this field with the value provided in the URL integrated request parameter |
| includeTag | Yes | true or false | Fill this field with the value provided in the URL Tag request parameter |
| creatorBA | Yes | PJM | Refer to section 2.2. |
| requestorBA | Yes | MISO | Refer to section 2.3. |

5.2. Real-Time NSI

Description:

This is the real-time NSI checkout required by NERC INT-009.

Data Elements:

| Field Name | Required | Sample Value | Description |
|-------------------|----------|--------------------------|--|
| checkoutBA | Yes | MISO | Refer to section 2.4. |
| intervalStartTime | Yes | 2015-05-15T21:00:00.000Z | Fill this field with start time of the interval. |
| intervalStopTime | Yes | 2015-05-15T21:15:00.000Z | Fill this field with stop time of the interval. |
| sinkBA | Yes | PJM | Refer to section 2.2. |
| mwNet | Yes | 2 | Refer to section 2.6 |
| verifiedMatch | Yes | true | Refer to section 5.6. |
| overriddenFlag | No | true | Fill this field with true if a manual override has been applied to the NSI value. Else, false (default). |

5.3. Daily NSI

Description:

There are two times when BAs typically request each other's daily NSI totals.

After-the-Fact (ATF) NSI checkout targets *the previous operating day* and is conducted for the purpose of reconciling NSI numbers used in Energy Accounting tasks. BAs check the entire day's NSI total, and in the event the daily totals don't match between BAs, each hourly total can be checked for discrepancies.

Before-the-fact (BTF) NSI checkout targets *the next operating day* and is conducted for the purpose of verifying tomorrow's anticipated interchange schedule. Some BAs use this data along with their next day weather report to develop an expected interchange curve while others use it as an input to various look-ahead assessments that commit generation.

Data Elements:

| Field Name | Required | Sample Value | Description |
|-------------------|----------|--------------------------|--|
| checkoutBA | Yes | MISO | Refer to section 2.4. |
| intervalStartTime | Yes | 2015-05-15T04:00:00.000Z | Fill this field with start time of the interval. |
| intervalStopTime | Yes | 2015-05-16T04:00:00.000Z | Fill this field with stop time of the interval. |
| sinkBA | Yes | PJM | Refer to section 2.5. |
| mwDaily | Yes | 2 | Refer to section 2.6 |
| verifiedMatch | Yes | true | Refer to section 5.6. |

5.4. Optional Integrated Interval Data

Description:

If a request is made for NSI data with integrated data included, the payload shall include all hours contributing to the NSI total provided in payload.

Data Elements:

| Field Name | Required | Sample Value | Description |
|-------------------|----------|--------------------------|---|
| intervalStartTime | Yes | 2015-05-15T21:00:00.000Z | UTC |
| intervalStopTime | Yes | 2015-05-15T21:00:00.000Z | UTC |
| sinkBA | Yes | PJM | The Entity Code of the BA where net energy is flowing into. |
| mwNetIntegrated | Yes | 2 | Absolute value of the hourly integrated NSI MW |
| verifiedMatch | Yes | true | Refer to section 5.6. |

5.5. Optional eTag Data

Description:

If a request is made for NSI data with underlying eTag data included, the payload shall include all eTags contributing to the NSI total provided in payload.

Data Elements:

| Field Name | Required | Sample Value | Description |
|--------------------|----------|------------------------------|--|
| tagIndex | Yes | 123456 | Fill Tag index value in this field. |
| tagName | Yes | MISO_CRGL1ABDD 01_PJM | Fill Tag name value in this field. |
| tagTransactionType | Yes | Normal | Normal or Emergency are the only Tag types that are counted towards NSI. Dynamic and Pseudo-tie types are not counted towards NSI. Refer to section 2.6 for more details. |
| startTime | Yes | 2015-05- 15T21:00:00.000Z | Refer to profile building section to see how to fill this value. |
| endTime | Yes | 2015-05- 15T21:00:00.000Z | Refer to profile building section to see how to fill this value. |
| mwEnergy | Yes | 2 | Refer to profile building section to see how to fill this value. |
| tagUpdateTimestamp | Yes | 2015-05- 15T21:00:00.000Z | Fill this field with the last time the tag was updated to its current final state. Which means every time when existing tag gets curtailed/adjusted this timestamp should be updated. This helps counter party to know how latest is other party's data per tag. This also could help in programmatic tag discrepancies solutions. |

Profile Building for eTag

Single eTag profiles may be large depending on the number of requests created on each eTag. Users may choose to only include time periods that affect the NSI and compact the matching intervals.

Example

If a request parameter is from 2019-08-11T13:00:00.000Z to 2019-08-11T15:00:00.000Z, then this two hour period has 8 'typical' tagging intervals of 15 minutes each. All eTags contributing to these 8 intervals may have various time profiles. Examples of how Tag profile limits should be applied are provided in the table below.

| Field Name | Actual Tag profile | Suggested Tag profile for payload |
|------------|--|--|
| Tag 1 | From : 2019-08-10T04:00:00.000Z To : 2019-08-12T04:00:00.000Z MW: 50 | From : 2019-08-10T04:00:00.000Z To : 2019-08-12T04:00:00.000Z MW: 50 |
| Tag 2 | From : 2019-08-11T13:30:00.000Z To : 2019-08-11T14:30:00.000Z MW: 50 | From : 2019-08-11T13:30:00.000Z To : 2019-08-11T14:30:00.000Z MW: 50 |
| Tag 3 | From : 2019-08-11T12:00:00.000Z To : 2019-08-11T13:00:00.000Z MW: 50 From : 2019-08-11T13:00:00.000Z To : 2019-08-11T14:00:00.000Z MW: 60 From : 2019-08-11T14:00:00.000Z To : 2019-08-11T15:00:00.000Z MW: 70 | From : 2019-08-11T13:00:00.000Z To : 2019-08-11T14:00:00.000Z MW: 60 From : 2019-08-11T14:00:00.000Z To : 2019-08-11T15:00:00.000Z MW: 70 |

| Field Name | Actual Tag profile | Suggested Tag profile for payload |
|------------|--|--|
| | From : 2019-08-11T15:00:00.000Z To : 2019-08-11T16:00:00.000Z MW: 80 | |
| Tag 4 | From : 2019-08-11T11:30:00.000Z To : 2019-08-11T12:30:00.000Z MW: 50 From : 2019-08-11T12:30:00.000Z To : 2019-08-11T13:30:00.000Z MW: 60 From : 2019-08-11T13:30:00.000Z To : 2019-08-11T14:30:00.000Z MW: 70 From : 2019-08-11T14:30:00.000Z To : 2019-08-11T15:30:00.000Z MW: 80 From : 2019-08-11T15:30:00.000Z To : 2019-08-11T16:30:00.000Z MW: 90 | From : 2019-08-11T12:30:00.000Z To : 2019-08-11T13:30:00.000Z MW: 60 From : 2019-08-11T13:30:00.000Z To : 2019-08-11T14:30:00.000Z MW: 70 From : 2019-08-11T14:30:00.000Z To : 2019-08-11T15:30:00.000Z MW: 80 |

6. Three-Part Verification

Description:

This section provides general guidance on how a BA might manage the verification and recording of automated checkout data. This approach adheres to the general principles outlined in the NERC COM-002 reliability standard to describe the effective use of the `verifiedMatch` flag.

AS BAs attempt to automate the checkout process, they need to keep records to ensure the requesting party both received the creator's payload and confirmed the provided NSI values. Incorporating the `<verifiedMatch>false</verifiedMatch>` element into the payload helps to ensure that BAs perform this three-part verification process.

State Management:

Below are important pieces of information a BA will need to record and maintain for each interval in order to be able to successfully verify NSI values with a neighbor.

1. **Own NSI** – each BA needs to maintain its own NSI value.
2. **Neighbor NSI** – each BA needs to record its neighbor's most recently received NSI value for use in the verification.
3. **Sink BA** – Each BA needs to record the direction of the power/interchange flow from their perspective and both BAs need to ensure that they agree upon the direction of the flow. In a BA's local database, the Sink BA can be translated into a flow direction as follows:
 - Create an additional data column that indicates the direction of interchange flow (import/export), *or*
 - Apply positive and negative signs to the NSI values
4. **Own Verified Flag** – Each BA needs to maintain its own verified flag. A BA should change this flag to 'true' when Own NSI and the Neighbor NSI are equivalent in magnitude and direction for a given time interval.
5. **Neighbor Verified Flag** – Each BA needs to record and maintain their neighbor's `verifiedMatch` flag. This confirms that the neighbor has also requested NSI data and has verified that the response matches their records.

Example:

The following is a sequence of events where everything correctly aligns. This example shows how data would look within a two BAs' systems where BA1 is the requestor BA exchanging data with a neighbor (BA2), who is the creator BA.

1. Within each BA's system, a Verified Flag is maintained for both the BA and their neighbor. When there is no Tag data, each NSI column shall have *null* for NSI and each Verified Flag shall be set to 'F'.

BA1's System

| Event | BA1 NSI | Neighbor NSI | BA1 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 0 | Null | Null | F | F |

BA2's System

| Event | BA2 NSI | Neighbor NSI | BA2 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 0 | Null | Null | F | F |

2. When a confirmed Tag comes into BA1's system, and they have not yet requested their neighbor's data, BA1 will record its own NSI value and leave the Verified Flag set to 'F'.

BA1's System

| Event | BA1 NSI | Neighbor NSI | BA1 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| | | | | |

| | | | | |
|---|-----|------|---|---|
| 1 | 100 | Null | F | F |
|---|-----|------|---|---|

Likewise, when a confirmed Tag comes into BA2's system. They, too, record their own NSI value and leave their Verified Flag set to 'F'.

BA2's System

| Event | BA2 NSI | Neighbor NSI | BA2 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 1 | 100 | Null | F | F |

- At a designated time, BA1 initiates a request (*BA1 is the 'requestor'*) from the neighbor BA (BA2) specified on the Tag. The neighbor then sends a response (*BA2 is the 'creator'*) back to BA1 where it is received.
 - If the NSI data between BA1 and the neighbor match, the BA1 Verified Flag is set to 'T'.
 - If the NSI data does not match, the BA1 Verified Flag remains set to 'F'.

BA1's System

| Event | BA1 NSI | Neighbor NSI | BA1 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 2 | 100 | 100 | T | F |

At this point BA2 has not requested BA1's data yet. Therefore, BA2's data remains unchanged within their system.

BA2's System

| Event | BA2 NSI | Neighbor NSI | BA2 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 2 | 100 | Null | F | F |

- At a designated time, BA2 initiates a request (*BA2 is the 'requestor'*) to BA1. BA1 then sends a response (*BA1 is the 'creator'*) back to BA2 where it is received. The response payload contains the BA1 Verified Flag value by setting the verifiedMatch element as "true".

BA1's System

| Event | BA1 NSI | Neighbor NSI | BA1 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 3 | 100 | 100 | T | F |

BA2's System

| Event | BA2 NSI | Neighbor NSI | BA2 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 3 | 100 | 100 | T | T |

- BA1 initiates another request from BA2, who then sends a response with the verifiedMatch element set to "true". BA1 shall then set the Neighbor Verified Flag to 'T'.

BA1's System

| Event | BA1 NSI | Neighbor NSI | BA1 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 4 | 100 | 100 | T | T |

BA2's System

| Event | BA2 NSI | Neighbor NSI | BA2 Verified Flag | Neighbor Verified Flag |
|-------|---------|--------------|-------------------|------------------------|
| 4 | 100 | 100 | T | T |



At this point, each BA shall have both Verified Flags (own & neighbor) set to 'T', which indicates both parties have independently verified the NSI totals. This state can be considered as 'checked out'.

7. XSD Definitions

7.1. Schema

```

<?xml version="1.0" encoding="utf-8"?>
<!-- edited with XMLSpy v2020 sp1 (x64) (http://www.altova.com) by PJM (PJM INTERCONNECTION, LLC) -->
<xs:schema xmlns:nsi="http://www.pjm.com/external/schemas/nsi/v1" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.pjm.com/external/schemas/nsi/v1">
  <xs:element name="NsiCheckout" type="nsi:NsiCheckout"/>
  <xs:complexType name="NsiCheckout">
    <xs:sequence>
      <xs:element name="requestStartTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
      <xs:element name="requestStopTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
      <xs:element name="responseTimestamp" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
      <xs:element name="requestType" type="nsi:RequestType" minOccurs="1" maxOccurs="1"/>
      <xs:element name="includeIntegrated" type="xs:boolean" default="false" minOccurs="1" maxOccurs="1"/>
      <xs:element name="includeTag" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
      <xs:element name="creatorBA" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="RequestorBAs" type="nsi:RequestorBAs" minOccurs="0" maxOccurs="1"/>
      <xs:choice minOccurs="1" maxOccurs="1">
        <xs:element ref="nsi:NsiTotals" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="nsi:DailyNsiTotals" minOccurs="1" maxOccurs="1"/>
      </xs:choice>
      <xs:element name="RealTimeEnergyTransactions" type="nsi:RealTimeEnergyTransactions" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="RequestType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="DAY"/>
      <xs:enumeration value="RT"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="NsiTotals" type="nsi:NsiTotals"/>
  <xs:complexType name="NsiTotals">
    <xs:sequence>
      <xs:element name="NsiTotal" type="nsi:NsiTotal" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="RealTimeEnergyTransactions" type="nsi:RealTimeEnergyTransactions"/>
  <xs:complexType name="RealTimeEnergyTransactions">
    <xs:sequence>
      <xs:element name="RealTimeEnergyTransaction" type="nsi:RealTimeEnergyTransaction" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="DailyNsiTotals" type="nsi:DailyNsiTotals"/>
  <xs:complexType name="DailyNsiTotals">
    <xs:sequence>
      <xs:element name="DailyNsiTotal" type="nsi:DailyNsiTotal" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="NsiIntervals" type="nsi:NsiIntervals"/>
  <xs:complexType name="NsiIntervals">
    <xs:sequence>
      <xs:element name="NsiInterval" type="nsi:NsiInterval" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="NsiInterval" type="nsi:NsiInterval"/>
  <xs:complexType name="NsiInterval">
    <xs:sequence>

```

```

<xs:element name="intervalStartTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
<xs:element name="intervalStopTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
<xs:element name="sinkBA" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="mwNet" type="xs:integer" minOccurs="1" maxOccurs="1"/>
<xs:element name="verifiedMatch" type="xs:boolean" default="false" minOccurs="1" maxOccurs="1"/>
<xs:element name="overriddenFlag" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
<xs:element name="DailyNsiTotal" type="nsi:DailyNsiTotal"/>
<xs:complexType name="DailyNsiTotal">
  <xs:sequence>
    <xs:element name="checkoutBA" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element ref="nsi:DailyNsiIntervals" minOccurs="1" maxOccurs="1"/>
    <xs:element ref="nsi:IntegratedIntervals" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="DailyNsiIntervals" type="nsi:DailyNsiIntervals"/>
<xs:complexType name="DailyNsiIntervals">
  <xs:sequence>
    <xs:element name="DailyNsiInterval" type="nsi:DailyNsiInterval" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="RealTimeEnergyTransaction" type="nsi:RealTimeEnergyTransaction"/>
<xs:complexType name="RealTimeEnergyTransaction">
  <xs:sequence>
    <xs:element name="tagIndex" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="tagName" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="tagTransactionType" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="tagUpdateTimestamp" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <xs:element ref="nsi:Profiles" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Profiles" type="nsi:Profiles"/>
<xs:complexType name="Profiles">
  <xs:sequence>
    <xs:element name="Profile" type="nsi:Profile" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="DailyNsiInterval" type="nsi:DailyNsiInterval"/>
<xs:complexType name="DailyNsiInterval">
  <xs:sequence>
    <xs:element name="intervalStartTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <xs:element name="intervalStopTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <xs:element name="sinkBA" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="mwDaily" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="verifiedMatch" type="xs:boolean" default="false" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Profile" type="nsi:Profile"/>
<xs:complexType name="Profile">
  <xs:sequence>
    <xs:element name="startTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <xs:element name="endTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <xs:element name="mwEnergy" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="IntegratedIntervals" type="nsi:IntegratedIntervals"/>
<xs:complexType name="IntegratedIntervals">
  <xs:sequence>
    <xs:element name="IntegratedInterval" type="nsi:IntegratedInterval" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

```

</xs:complexType>
<xs:element name="IntegratedInterval" type="nsi:IntegratedInterval"/>
<xs:complexType name="IntegratedInterval">
  <xs:sequence>
    <xs:element name="intervalStartTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <xs:element name="intervalStopTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    <xs:element name="sinkBA" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="mwNetIntegrated" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="verifiedMatch" type="xs:boolean" default="false" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="RequestorBAs" type="nsi:RequestorBAs"/>
<xs:complexType name="RequestorBAs">
  <xs:sequence>
    <xs:element name="requestorBA" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="NsiTotal" type="nsi:NsiTotal"/>
<xs:complexType name="NsiTotal">
  <xs:sequence>
    <xs:element name="checkoutBA" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element ref="nsi:NsiIntervals" minOccurs="1" maxOccurs="1"/>
    <xs:element ref="nsi:IntegratedIntervals" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
    
```

7.2. Sample XML

Description:

Below are examples of well-formed xml.

Example 1

```

<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2020 sp1 (x64) (http://www.altova.com)-->
<nsi:NsiCheckout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:nsi="http://www.pjm.com/external/schemas/nsi/v1"
xsi:schemaLocation="http://www.pjm.com/external/schemas/nsi/v1 nsi.xsd">
  <requestStartTime>2001-12-17T09:30:47Z</requestStartTime>
  <requestStopTime>2001-12-17T09:30:47Z</requestStopTime>
  <responseTimestamp>2001-12-17T09:30:47Z</responseTimestamp>
  <requestType>RT</requestType>
  <includeIntegrated>true</includeIntegrated>
  <includeTag>true</includeTag>
  <creatorBA>String</creatorBA>
  <RequestorBAs>
    <requestorBA>String</requestorBA>
  </RequestorBAs>
  <nsi:NsiTotals>
    <NsiTotal>
      <checkoutBA>String</checkoutBA>
      <nsi:NsiIntervals>
        <NsiInterval>
          <intervalStartTime>2001-12-17T09:30:47Z</intervalStartTime>
          <intervalStopTime>2001-12-17T09:30:47Z</intervalStopTime>
          <sinkBA>String</sinkBA>
          <mwNet>0</mwNet>
          <verifiedMatch>>false</verifiedMatch>
          <overriddenFlag>>false</overriddenFlag>
        </NsiInterval>
      </nsi:NsiIntervals>
    </NsiTotal>
  </nsi:NsiTotals>
</nsi:NsiCheckout>
    
```

```

    </NsiInterval>
  </nsi:NsiIntervals>
  <nsi:IntegratedIntervals>
    <IntegratedInterval>
      <intervalStartTime>2001-12-17T09:30:47Z</intervalStartTime>
      <intervalStopTime>2001-12-17T09:30:47Z</intervalStopTime>
      <sinkBA>String</sinkBA>
      <mwNetIntegrated>0</mwNetIntegrated>
      <verifiedMatch>>false</verifiedMatch>
    </IntegratedInterval>
  </nsi:IntegratedIntervals>
</NsiTotal>
</nsi:NsiTotals>
<RealTimeEnergyTransactions>
  <RealTimeEnergyTransaction>
    <tagIndex>0</tagIndex>
    <tagName>String</tagName>
    <tagTransactionType>String</tagTransactionType>
    <tagUpdateTimestamp>2001-12-17T09:30:47Z</tagUpdateTimestamp>
  <nsi:Profiles>
    <Profile>
      <startTime>2001-12-17T09:30:47Z</startTime>
      <endTime>2001-12-17T09:30:47Z</endTime>
      <mwEnergy>0</mwEnergy>
    </Profile>
  </nsi:Profiles>
</RealTimeEnergyTransaction>
</RealTimeEnergyTransactions>
</nsi:NsiCheckout>

```

Example 2

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Sample XML file generated by XMLSpy v2020 sp1 (x64) (http://www.altova.com)-->
<nsi:NsiCheckout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:nsi="http://www.pjm.com/external/schemas/nsi/v1"
xsi:schemaLocation="http://www.pjm.com/external/schemas/nsi/v1 nsi.xsd">
  <requestStartTime>2001-12-17T09:30:47Z</requestStartTime>
  <requestStopTime>2001-12-17T09:30:47Z</requestStopTime>
  <responseTimestamp>2001-12-17T09:30:47Z</responseTimestamp>
  <requestType>DAY</requestType>
  <includeIntegrated>>true</includeIntegrated>
  <includeTag>true</includeTag>
  <creatorBA>String</creatorBA>
  <RequestorBAs>
    <requestorBA>String</requestorBA>
  </RequestorBAs>
  <nsi:DailyNsiTotals>
    <DailyNsiTotal>
      <checkoutBA>String</checkoutBA>
    </DailyNsiTotal>
  </nsi:DailyNsiTotals>
  <nsi:DailyNsiIntervals>
    <DailyNsiInterval>
      <intervalStartTime>2001-12-17T09:30:47Z</intervalStartTime>
      <intervalStopTime>2001-12-17T09:30:47Z</intervalStopTime>
      <sinkBA>String</sinkBA>
      <mwDaily>0</mwDaily>
      <verifiedMatch>>false</verifiedMatch>
    </DailyNsiInterval>
  </nsi:DailyNsiIntervals>
  <nsi:IntegratedIntervals>
    <IntegratedInterval>
      <intervalStartTime>2001-12-17T09:30:47Z</intervalStartTime>

```

```
<intervalStopTime>2001-12-17T09:30:47Z</intervalStopTime>
<sinkBA>String</sinkBA>
<mwNetIntegrated>0</mwNetIntegrated>
<verifiedMatch>>false</verifiedMatch>
</IntegratedInterval>
</nsi:IntegratedIntervals>
</DailyNsiTotal>
</nsi:DailyNsiTotals>
<RealTimeEnergyTransactions>
  <RealTimeEnergyTransaction>
    <tagIndex>0</tagIndex>
    <tagName>String</tagName>
    <tagTransactionType>String</tagTransactionType>
    <tagUpdateTimestamp>2001-12-17T09:30:47Z</tagUpdateTimestamp>
    <nsi:Profiles>
      <Profile>
        <startTime>2001-12-17T09:30:47Z</startTime>
        <endTime>2001-12-17T09:30:47Z</endTime>
        <mwEnergy>0</mwEnergy>
      </Profile>
    </nsi:Profiles>
  </RealTimeEnergyTransaction>
</RealTimeEnergyTransactions>
</nsi:NsiCheckout>
```



8. Terms of Use

Please refer to PJM NSI Automated Checkout Data Specification Terms of Use at <https://www.pjm.com/markets-and-operations/etools/exschedule/nsi-automated-checkout.aspx>.